Решающие деревья. Градиентный бустинг. Оценка модели.

Гончаров Павел kaliostrogoblin3@gmail.com
Нестереня Игорь nesterione@gmail.com

План занятия

- Повторение
- Алгоритмы основанные на решающих деревьях
- Ансамбли алгоритмов. Случайный лес. Градиентный бустинг
- Метрики оценки моделей

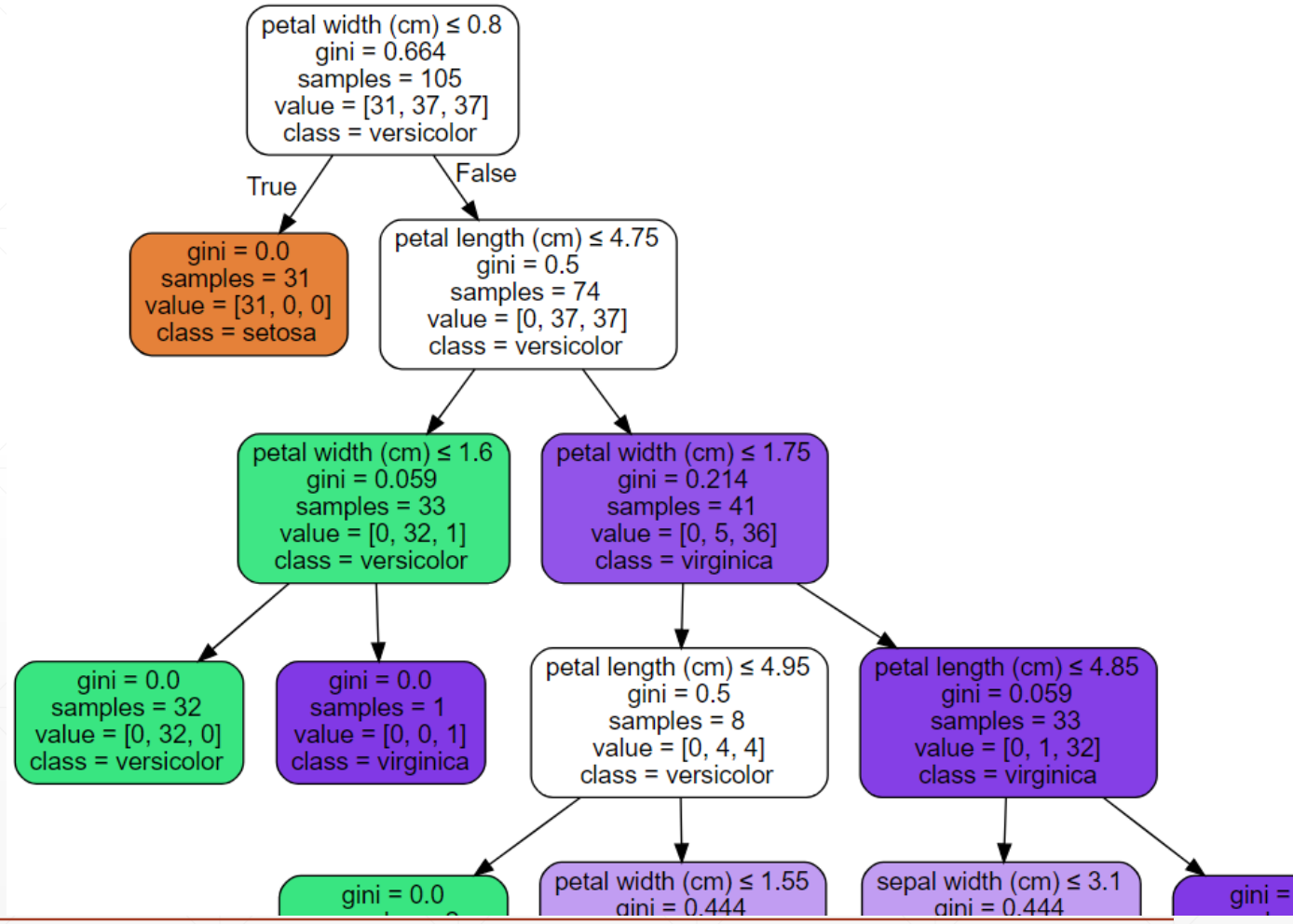
Повторение

- Какие алгоритмы изучили?
- Постановка задачи. Входные/выходные параметры алгоритма?
- Функции потерь, целевая функция?
- Методы многомерной оптимизации?
- Нормализация данных?
- Переобучение и регуляризация?

Решающие деревья

- Класс методов машинного обучения
- Для классификации (есть обобщение для регрессии, CART деревья)

Алгоритм представляет собой бинарное дерево, с предикатами (вопросами) в узлах и меткой класса в листьях.



Обучение

- Поиск оптимальных пороговых параметров или оптимальных дихотомических разбиений для признаков x1… xn.
- Поиск производится с целью снижения индекса неоднородности в выборках.
- Индекс вычисляется для любой произвольной подвыборки.
    - Энтропийный индекс неоднородности
    - Индекс Джини $I(S) = 1 - \sum_i P_i^2$, где Pi - доля объектов класса Ki в выборке S
    - Индекс ошибочной классификации

Обучение

- Функция LearnID3
  - Если в одном классе, вернуть лист
- Найти предикат с максимальной **информативностью** I (индекса неоднородности)
  - Разбить на 2 части
  - Если разбиения не было, создать лист
  - Рекурсивно вызываем для каждой части

Извлечение значимых признаков

Преимущества

- Быстро обучается
- Гибкий алгоритм
- Интерпретируемый результат
- Хорошо работает на данных с пропусками
  - Объект отправляется в оба поддерева, ответы усредняются с весами

Недостатки

- Жадный
- Высокая чувствительность к шуму
- Проблема переусложнения

Способы устранения недостатков

- Редукция
  - Подрезание (pruning) Алгоритм C4.5
  - Cost-complexity pruning
- Композиция

Композиция алгоритмов

- h(x)=$sing\sum_{t=1} b_t(x)$ - голосование классификаторов

- обучение по случайным подвыборкам
- обучение по выборке со случайными весами объектов
- обучение по случайным подмножествам признаков
- использование различных моделей классификации
- …

Беггинг

- Беггинг и случайный лес (bootstrap aggregation) – обучается по т классификаторов на случайных подвыборках
- Метод случайных подпространств (random space method) – обучается по случайным подмножествам n’
- можно совмещать

Случайный лес (Random forest)

- беггинг над решающими деревьями
- признак в каждой вершине выбирается из случайного подмножества k признаков

Градиентный бустинг (Gradient Boosting)

- взвешенное голосование
- независим от функции потерь
- строит алгоритмы друг за другом (улучшая друг предыдущие)

$$H(x) = \sum_t \alpha_t h_t\ (x)$$

Алгоритм

- вход X, T, где T – количество классификаторов
- выход базовые алгоритмы и их весе $\propto$
- для всех T
  - найти базовый алгоритм
  - решить задачу одномерной оптимизации, найти $\propto$
  - включить значения в композицию на объектах выборки

Меры качества

Меры качества

- Accuracy $\frac{1}{N}\sum_{i=1}[a(x_i) = y_i]$
- Проблема несбалансированных выборок (Например 10 одного класса, 990 другого класса)

- Смотреть базовую долю правильных ответов

Типы ошибок

	y=1	y=1
h(x)=1	True Positive (TP)	False Positive (FP)
h(x)=0	False Negative (FN)	True Negative (TN)

$$\text{accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Точность и полнота

- precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{TP+FN}$

- чем выше точность, тем меньше ложных срабатываний
- чем выше полнота, тем ниже ложных пропусков

- Гармоническое среднее или F-мера = $\frac{2*precision*recall}{precision+recall}$

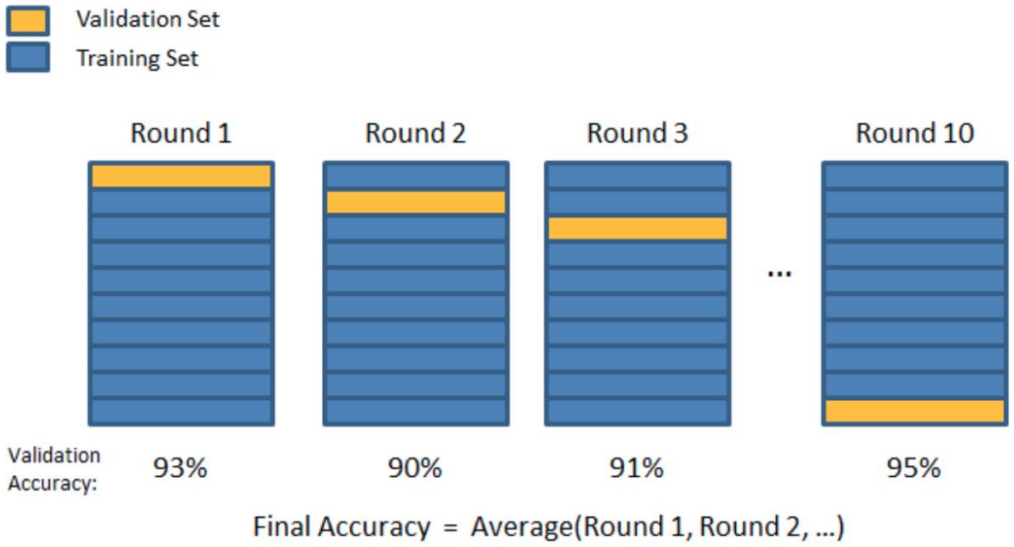
Техники подбора модели

- Основная техника кросс–валидация, или скользящий контроль.

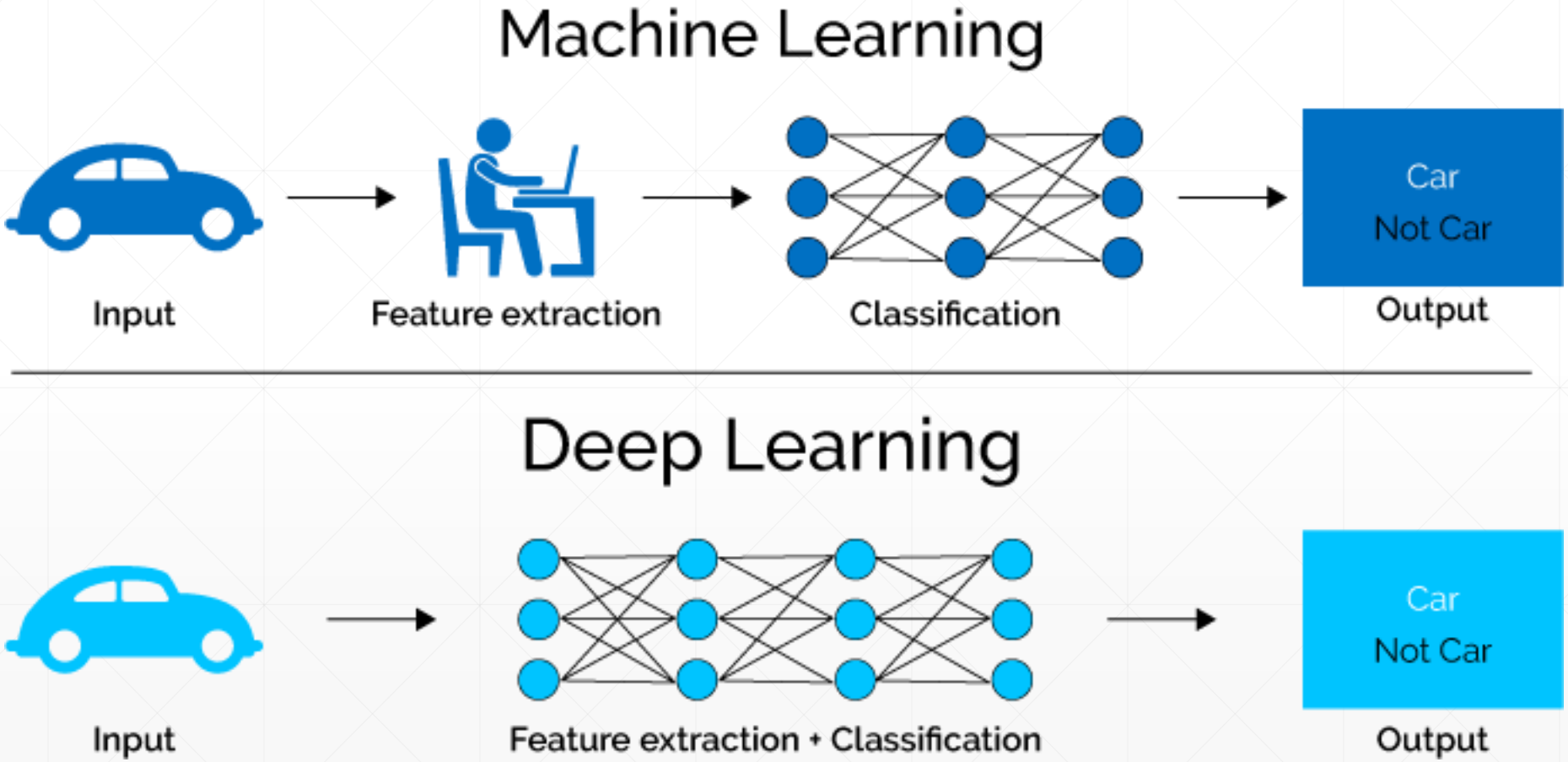
Для подбора модели обычно используют Grid search

K-fold

- k-fold кросс-валидация разбивает исходную выборку на k равных по размеру подвыборки. Один набор используется для валидации, и k-1 для обучения. Кросс-валидация повторяется k раз для каждой подвыборки. Потом результат усредняется.



ML -> DL


Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output
Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output